

qRULES v8.1 USER GUIDE

PRODUCT: qRules v8.1
LAST UPDATED: June 5, 2017

VALUE PROPOSITION

InfoPath is the best forms editor on the market today because it lets you create rich forms with structured data without having to rely on a developer to code features. Business process owners can quickly prototype and publish forms with no dependency on developers or IT staff to deploy the forms. Less code means less cost to you both in up front development work and long-term maintenance work. You quickly create a form and when changes are needed, you can do them yourself.

However, many simple actions in InfoPath require code, like sorting tables or checking to see if the form is new. The result is that today most forms need a small amount of code to do a few simple things. That's really too bad because it creates a barrier for most people using InfoPath.

Qdabra has created a common library of code snippets to cover these "codeless" gaps. We call this library qRules. qRules has been around since 2009 and contains commonly requested functions accessible via rules instead of code. We've even extended qRules to do things that aren't available in InfoPath today – like encrypting fields.

REQUIREMENTS

- qRules requires InfoPath 2007 SP2 or newer.
- qRules Injector requires .NET Framework 4.0 or greater.
- qRules injected forms require .NET Framework 4.0 or greater.
- qRules is mainly intended for use in forms developed off the blank, new form template type. qRules is not compatible with SharePoint List forms because these don't allow code. Other form types, such as email forms or Document Information Panel forms, may support qRules, but in depth testing has not been performed.
- Additional requirements for individual commands will be discussed in command-specific documentation.
- qRules is code, and is under the same constraints as any form with code in terms of browser compatibility. The deprecation of the Sandbox in Office 365 means qRules forms may no longer be browser compatible in an Office 365 environment. You can learn more [here](http://www.qdabra.com).



TABLE OF CONTENTS

Resources.....	3
Hints and Tips	4
QdabraRules Data Source Nodes	13
Create a Query Data Connection to DBXLDocumentService Submit Document Method	17
Install QdFile	19
Data Structure for Picture and File Attachment Nodes	21
Using qRules in InfoPath Forms Services (IPFS)	24
Remove qRules From a Form Template	34
Change Log	37
Support	51



RESOURCES

If you have not done so already, we suggest your qRules journey begin with the **qRules Installation Guide**, included alongside the qRules installer.

The **qRules Quick Start** document will walk you through injecting an InfoPath form and setting up one command. This will help illustrate the basic usage of qRules.

You could then:

- Check out Qdabra's qRules training modules and videos, which are available by clicking [here](#).
- Refer to the **qRules Command Overview**. This document includes information on all commands included in qRules.
- Try the **qRulesForm.xsn** to see command samples. In default installations, this form is included in the following folder: *C:\Program Files\Qdabra Software\Qdabra Rules Library*. To use this form, navigate to this folder, right-click on qRulesForm.xsn and select **Design**. Make sure that you select the default view, called **Main**, and click **Preview**.
- Refer to the **qRules User Guide** for various hints and tips as you work with Qdabra qRules.
- **qRules Full Trust Form Installation Guide**: Explains how to install the "Qdabra InfoPath to SharePoint List Tool". In default installations, you'll find this in *C:\Program Files\Qdabra Software\Qdabra Rules Library\Forms\Installable Forms*.

Please refer to the qRules product page on Qdabra.com for an updated list of all available documentation.

INJECTOR OPTIONS

The qRules injector has a number of options available. Generally, use the version of InfoPath that matches the compatibility of your template, and, unless needing rich client only commands, the Sandbox option. Only a few commands require using the Admin Deploy injection option – even if you intend to admin deploy your form.

Those are:

- SubmitAsUser
- QueryData, if the **ssapid** parameter is used
- RenderForm, if used in the browser
- SubmitToSharePointList, if the **ssapid** parameter is used

Please review the Command Compatibility Matrix at the end of the Command Overview for additional command requirements.



HINTS AND TIPS

qRules Sample Form

Every installation of qRules includes a sample form that illustrates the usage of some of the commands. This form can be found in the qRules Start Menu folder, within the qRules Forms shortcut.

This shortcut takes you to **C:\Program Files\Qdabra Software\Qdabra Rules Library\Forms** (if qRules was installed to the default location). Within this folder you'll find instructions ("qRules Full Trust Form Installation Guide.pdf") on how to install this form.

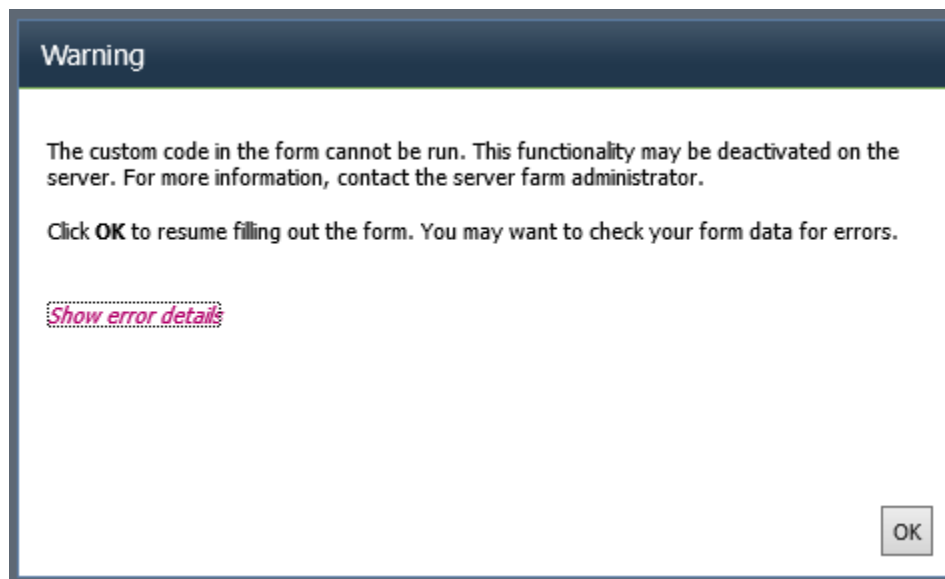
Though this form includes many of the qRules commands, some are too complex to illustrate in a simple fashion in this sample form. If you cannot find a command in the sample form, nor in the qRules [documentation](#) page, please contact [Qdabra Support](#).

Using qRules commands on Form Submit rules

In an InfoPath form template, you can change the Submit Options to "Perform custom action using rules". One can then add rules that will execute when a user clicks Submit from the Ribbon. Though qRules commands on the Form Submit rules execute in the Client, the same commands will fail to execute in the browser. As a workaround, Qdabra recommends removing the Ribbon (in Form Options) and instead using buttons that you have added to the view.

SharePoint 2010 SP2

After installing SP2 for SharePoint 2010, you may find that your qRules forms (and forms with code in general) stop working in the browser. The error encountered would look like this:



If you click **Show error details**, a correlation ID will show, and this will lead you to the following error in the SharePoint logs:

Could not load type 'Microsoft.SharePoint.Administration.SPService' from assembly 'Microsoft.SharePoint, Version=14.900.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c'.

To fix the issue, apply the October 2013 CU. If you need assistance to fix this issue, please contact Qdabra support.

Extend Sandbox timeout using Powershell scripts

In SharePoint 2010, the Microsoft SharePoint Foundation Sandboxed Code Service allows users to publish InfoPath forms that contain code directly to a form library. Sometimes, however, publishing may time out. The default timeout for the Microsoft SharePoint Foundation Sandboxed Code Service can be extended using PowerShell scripts.

```
#
# THESE LIMITS SHOULD ALLOW TO PUBLISH YOUR FORM
#
$spCodeService = [Microsoft.SharePoint.Administration.SPUserCodeService]::Local

$cpu = $spCodeService.ResourceMeasures["CPUExecutionTime"]

# NOTE THE VALUE BEFORE THE UPDATE TO RESET LATER
$cpu.AbsoluteLimit

$cpu.AbsoluteLimit = 600
$cpu.Update()

$cycles = $spCodeService.ResourceMeasures["ProcessCPUCycles"]

# NOTE THE VALUE BEFORE THE UPDATE TO RESET LATER
$cycles.AbsoluteLimit
$cycles.AbsoluteLimit = 2000000000000
$cycles.Update()

# NOTE THE VALUE BEFORE THE UPDATE TO RESET LATER
$spCodeService.WorkerProcessExecutionTimeout
$spCodeService.WorkerProcessExecutionTimeout = 600
$spCodeService.Update()

#
# "SharePoint 2010 User Code Host" must be restarted in the services applet
#
```

To set the timeout back to the default:



```
#
# RESET LIMITS TO DESIRED DEFAULTS
#
$spCodeService = [Microsoft.SharePoint.Administration.SPUserCodeService]::Local

$cpu = $spCodeService.ResourceMeasures["CPUExecutionTime"]
# RESET CPUExecutionTime to normal running level (60 is SharePoint default)
$cpu.AbsoluteLimit = 60
$cpu.Update()

$cycles = $spCodeService.ResourceMeasures["ProcessCPUCycles"]
# RESET ProcessCPUCycles to normal running level (100000000000 is SharePoint default)
$cycles.AbsoluteLimit = 100000000000
$cycles.Update()

# RESET WorkerProcessExecutionTimeout to normal running level (30 is SharePoint default)
$spCodeService.WorkerProcessExecutionTimeout = 30
$spCodeService.Update()

#
# "SharePoint 2010 User Code Host" must be restarted in the services applet
#
```

Cannot Preview a form with qRules?

After injecting an InfoPath form with qRules, you might be unable to open the XSN in the designer and then Preview. The error indicates:

InfoPath cannot open the selected form because of an error in the form's code.

Policy settings prevent opening Internet forms with managed code. To fix this problem, contact your system administrator.

We have seen this error:

- on machines that are not joined to a domain
- after performing an Internet Explorer upgrade, since these updates might reset your settings.

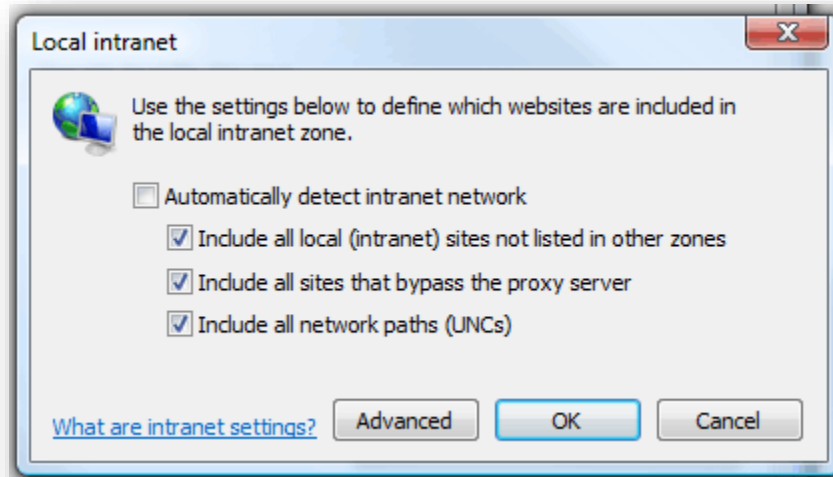
If you are seeing this error, there are two possible fixes to consider.

A. Modify the Internet Explorer security option causing the issue. Each user experiencing the issue will need to make the fix. The fix is to tell Internet Explorer that you are on an intranet. To do this:

1. Launch Internet Explorer.
2. Click the **Tools** button, and then click **Internet Options**.



3. On the **Security** tab of the **Internet Options** dialog box, select the **Local Intranet** zone, and then click **Sites**.
4. In the **Local Intranet** dialog box, clear the **Automatically Detect Intranet Network** check.
5. Select all other check boxes.



6. Click **OK** twice.
7. Preview your form.

B. If you are unable to use the solution above, specify a preview domain in the Form Options dialog box of your form template. This forces InfoPath to believe that the form is being opened in the specified domain. The form must be set to Domain Trust for this option to work.

1. Launch Internet Explorer.
2. Click the **Tools** button, and then click **Internet Options**.
3. In the **Security** tab, select the **Local Intranet** zone, click **Sites**, and then click **Advanced**.
4. Add (or ensure that you've already added) the website to the domain you will use for previewing.
5. Click **Close**, and then click **OK** twice.
6. Open your InfoPath form in Design mode.
7. Choose **Form Options** from the **Tools** menu.
8. Select the **Preview** category.
9. In the **Domain** text box, type the website that you added to the Internet Explorer Local Intranet Websites.
10. Click **OK**.
11. Preview your form.



Upgrading

Forms injected with a previous version will continue to work because they have had the qRules DLL injected into them. This means **you are not required to upgrade your forms a new version of qRules unless you wish to use the new functionality**. To take advantage of the new commands or fixes, you'll need to re-inject qRules into your form. Re-injecting the form upgrades the version of qRules used in your form but does **not automatically update the commands**.

Email forms

InfoPath email forms CAN be injected using qRules, but not all qRules commands have been tested for this type of template. In some cases, the InfoPath Email form template may default to the 2003 Object Model; for a solution, please see [this forum thread](#).

If you wish to develop an email form using qRules, please test and save your work frequently, and contact Qdabra Support if you run into any issues.

Migrate a form from a trial to a non-expiring (purchased) version of qRules

To inject your forms with a purchased version of qRules:

1. If you ran install.js during installation, execute uninstall.js. For details, see page 3.
2. Go to **Control Panel > Add/Remove programs** and uninstall your qRules trial.
3. Install the qRules version that you have purchased.
4. (optional) Execute install.js. For details, see page 3.
5. Use the qRules injector to re-inject your form with the purchased version of qRules.

Does your form have custom code?

If you would like to use qRules with your own code in an InfoPath form, please refer to [this PDF document](#) for details.

Occasional SharePoint error when Activating forms via Central Admin

Occasionally you may encounter an error on SharePoint 2007, 2010 or 2013 when activating a form template to a site. Other actions (performed via the Central Admin site UI) may also fail, and usually the workaround is to perform the same action using SharePoint Management Shell (PowerShell)

To activate a form (including qRules admin deployed forms) via PowerShell, use the following PowerShell commandlet:

```
enable-spinfopathformtemplate -Identity IPFS2013BrowserFormAdmin -Site http://qshp2013
```



enable-spinfopathformtemplate -Identity <FormName> -Site http://<site>

To disable, use:

disable-spinfopathformtemplate -Identity IPFS2013BrowserFormAdmin -Site http://qshp2013

Forms that have qRules cannot call AD in the browser if sandboxed

qRules code includes a form loading event. At the time of the release of v4.1, there was an issue with the DBXL ADUserInfo Web Service's GetMyInfo method and forms running in the sandbox with a form Loading event. If there is a form Loading event in the code, it appears that the GetMyInfo method executes as the system account that the sandbox runs under, if the GetMyInfo method also executes on load. The same issue appears to occur on the change of the finishedLoading node in the Qdabra Rules data source to true after qRules has finished loading.

The work around would be to run the GetMyInfo method data connection after the form is fully loaded – on the click of a different button or the change of a field. More info available in [this blog post](#).

.NET Framework 3.5 or greater is required to use templates injected with qRules

If you try to open a form that has been injected with qRules, InfoPath might be unable to create a blank form due to a problem with .NET 3.5. The full error message reads as follows:

InfoPath cannot open the selected form because of an error in the form's code.

The required version of the Microsoft .NET Framework is not installed on your computer, or the InfoPath Primary Interop Assembly (PIA) is not registered. Use Add or Remove Programs in Control Panel to make sure that the required version of the Microsoft .NET Framework is installed. Or install it using Windows Update and run the Setup program again to confirm that the corresponding version of the .NET Programmability Support is installed, or contact your system administrator.

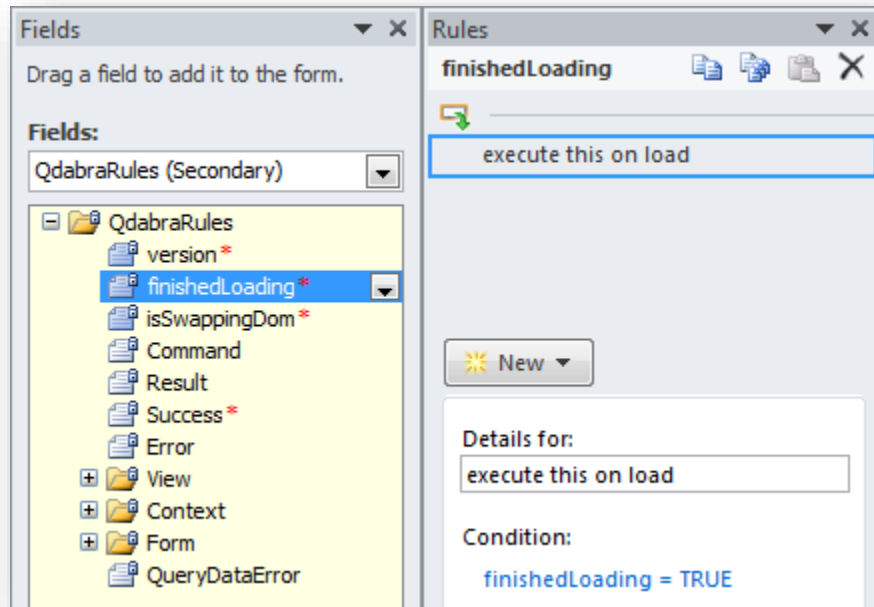
This message indicates that .NET is required. If you have already installed .NET 3.5 or greater and still receive the above error, then InfoPath was installed before .NET and the PIA has not been installed. To correct this, follow the instructions in the error message. That is, run a Repair using the Microsoft Office setup program to make sure that the .NET Programmability Support is installed.

Running rules on load

In InfoPath, rules run before code, and qRules commands are code based. This means that adding rules on load that execute qRules commands is not recommended. Instead, qRules has a node called **finishedLoading**. This node is Boolean, and it FALSE until the moment when the InfoPath form has



finished loading. At that moment the node becomes TRUE. To run qRules commands on load, we recommend adding a rule to the **finishedLoading** node, and adding a condition so the rule executes when the node is equal to TRUE.



Accidentally added code?

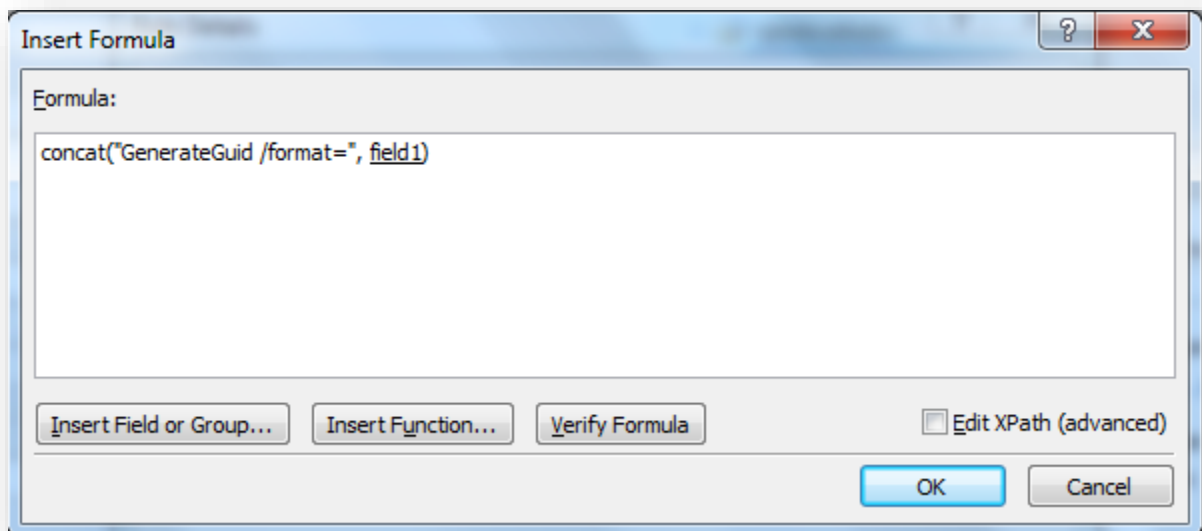
It is easy to accidentally add custom code to your form in InfoPath – unfortunately, that will disable qRules. If you find that qRules has been disabled in your form, you can do the following.

1. From Form Options / Programming click the Remove Code button
 - a. If Remove Code is disabled, add a loading event (for InfoPath 2007 – **Tools/Programming/Loading**; for InfoPath 2010 – **Developer** tab on the Ribbon and select **Loading Event**) and the Remove Code button should be enabled
 - b. Save and close the form
2. Reinject qRules, then try your command again.

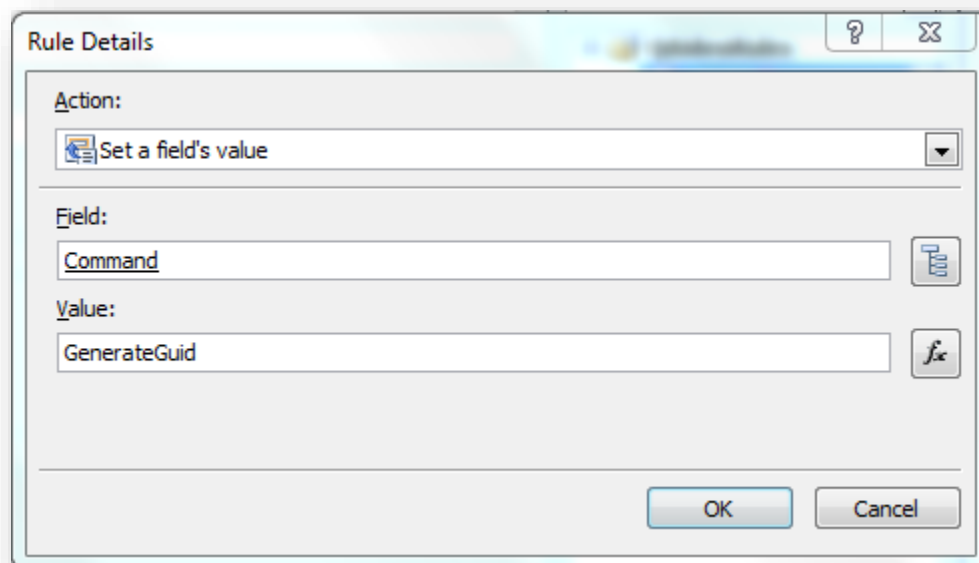
When to click fx?

When setting a field's value in InfoPath you'll notice an fx button. When setting the value of Command, you only need to enter the fx dialog if using concat, such as the example below.





If you are not using the value of a field in the form, then you do not need to Insert Formula. You can type the command into the Value box directly, as seen below.



Admin deployed forms in SharePoint produce multiple copies of Qdabra.Tools.RulesLibrary.dll in the Features folder

When a form is Admin deployed In SharePoint, it is expected that the DLL is put under the Features folder (C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\Template\Features\). Unfortunately this might lead to assembly conflicts, such as this:

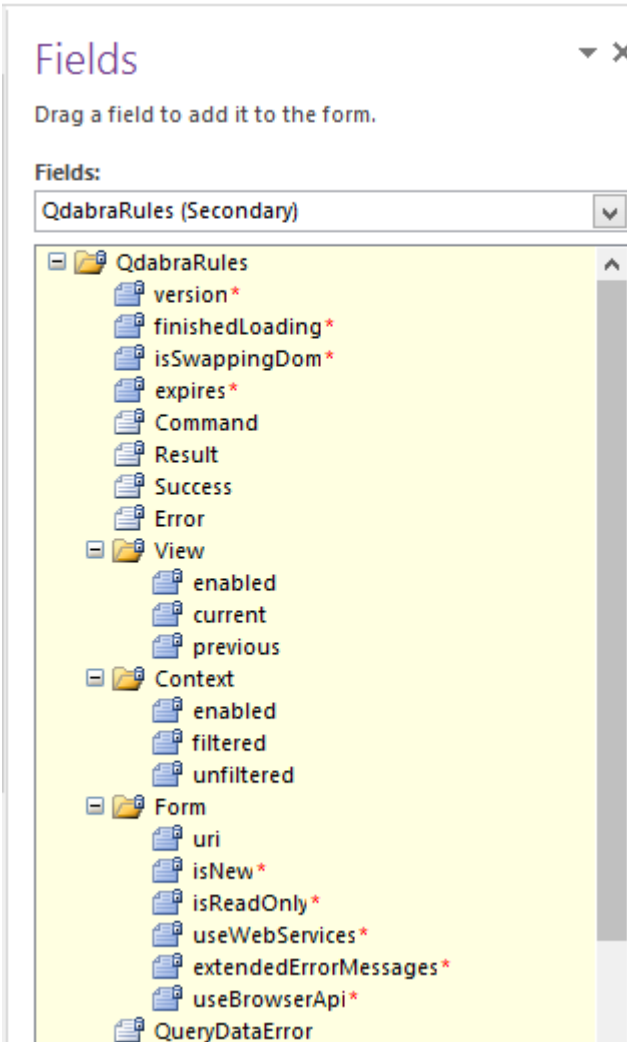
Exception Message: Could not load file or assembly 'file:///C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\Template\Features\FT-01-0c332152-5a92-4aad-eecd-d6979511e0af\Qdabra.Tools.RulesLibrary.dll' or one of its dependencies. The system cannot find the file specified.

If you encounter this or a similar error, please contact Qdabra support. One possible solution is to clean the Features folders to remove old versions, reinject the form and republish it. You may encounter this issue when using custom code DLLs or qRules in your forms (i.e. the issue is not limited to forms that contain qRules).



QDABRARULES DATA SOURCE NODES

There are a number of nodes in the QdabraRules data source. The following section describes the use and any defaults for some of these nodes.

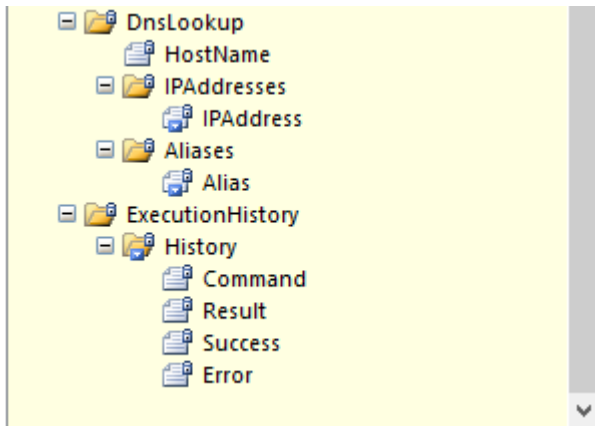


- **@version:** this field contains the current version of qRules
- **@finishedLoading:** qRules sets this field to true when the form finishes loading. You can use it to fire off rules that do things on load but after qRules has initialized. More information is available [here](#).
- **@isSwappingDom:** qRules uses this field in conjunction with the [SwapDomWithDocument](#) command to indicate when the data source (DOM) is being swapped. While this value is true, qRules disables updates to the form. If you are using the SwapDomWithDocument command and find rules are executing while the DOM swaps, you can use the value of this field in your rule conditions to ensure the rules only run if this field is false.
- **@expires:** if you are using the trial version of qRules, this field contains the expiration date (NOTE: preview the form to see the date value, it will not show in design mode).

- **Command:** this field serves as the access point for all qRules commands. Setting this field tells qRules to run the command specified in the value that you set.
- **Result:** after qRules executes the command, it places the result in this field (for commands that return a result). You can then add additional rules to set other fields to the value of the result field. You can also use the universal /resultxpath parameter to return the result to a form field.
- **Success:** this field is set to true if the command finishes execution with a successful status. If the command fails, this field is false.
- **Error:** when success is false, this field contains an error string describing what failed when qRules executed the command.



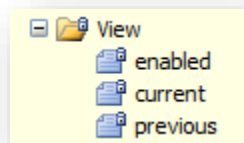
- **QueryDataError:** if an error is returned from a query while using the [QueryData](#) command, the error is returned to this node. Can be used for error handling query failures.



- **View, Context, Form and DnsLookup** groups are described in the sections below.
- **ExecutionHistory:** This group contains a repeating group called History, which stores every qRules Command executed (since the form was opened) along with each Result, Success and Error. More information in the section below.

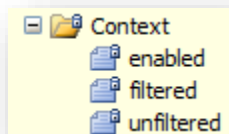
View

View tracking is disabled by default – setting *@enabled* in the View group to true will enable it. You can do this with a form load rule, or some other rule within your form logic – allowing you to only track views as needed. Once view tracking is enabled, the *@current* attribute will tell you the name of the current view and the *@previous* attribute will tell you the name of the previous view (if the user has switched views).



Context

The Context group identifies which node currently has focus.



Setting *@enabled* to true will enable this feature. Once Context tracking is enabled, *@filtered* and *@unfiltered* will contain the XPath of the focused field every time the user changes the focus. Values of *@filtered* and *@unfiltered* are generally the same, except for repeating data.



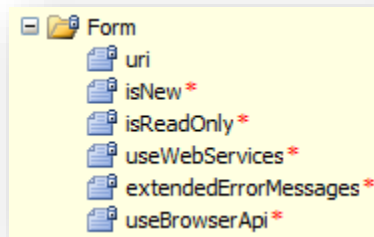
Note that *@filtered* and *@unfiltered* do not get their value set until the user changes the focus after *@enabled* is set to true.

It is a good practice to reset *@enabled* to false for better performance if you no longer need to track the focused field.

Context tracking is not available on browser forms.

Form

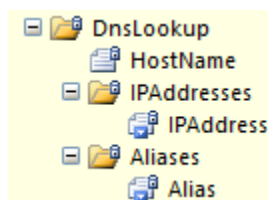
The Form group contains information about the form as well as other general settings and indicators.



- *@uri*: contains the template URI
- *@isNew*: a Boolean (true/false) that indicates whether the form is new.
- *@isReadOnly*: a Boolean that indicates whether the form is read only.
- *@useWebServices*: a Boolean that, when set to true, will use web services for SharePoint commands instead of the SharePoint Object Model in forms opened in the browser.
- *@extendedErrorMessages*: a Boolean that, when set to true, will return additional error information to the **Error** node
- *@useBrowserApi*: a Boolean that, when set to true, will force the form to always use the SharePoint Object Model for SharePoint commands.

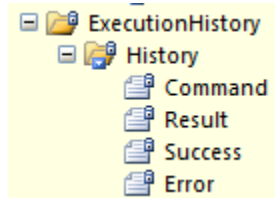
DnsLookup

Starting with qRules v5.2, there are two additional sections in the QdabraRules secondary data source. One of them is the DnsLookup group, which stores results returned by the LookupDns command.



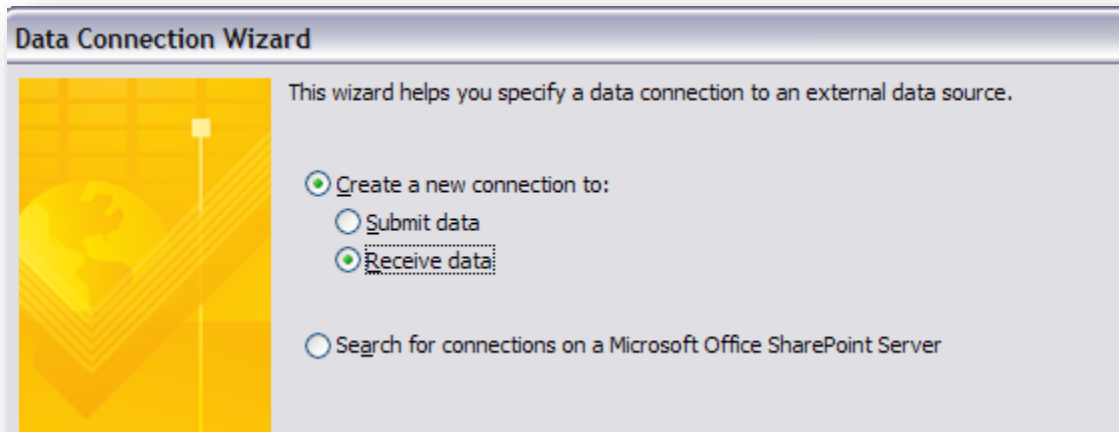
ExecutionHistory

Starting with qRules v5.2, there are two additional sections in the QdabraRules secondary data source. One of them is the ExecutionHistory group. The repeating table within this group stored a complete history of the commands executed by your InfoPath form, along with each Command's Result, Success and Error. This is invaluable for troubleshooting issues, and should be included in any debugging view or process. Simply drag these fields into your view, and display them as a repeating section or table.

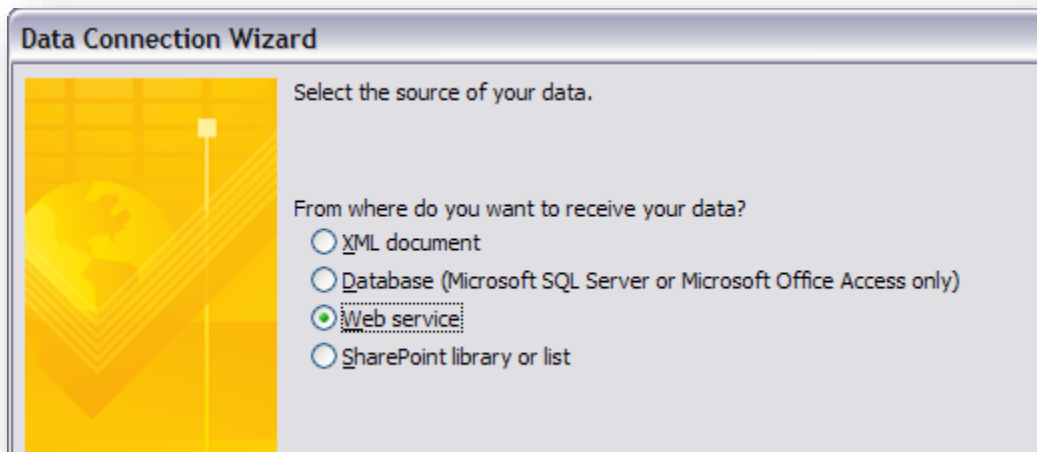


CREATE A QUERY DATA CONNECTION TO DBXLDOCUMENTSERVICE SUBMIT DOCUMENT METHOD

Some of the qRules commands require a *query* (receive) data connection to the Submit Document method of the DBXLDocumentService Web service. To create a query connection, start by adding a new data connection to your form that receives data:

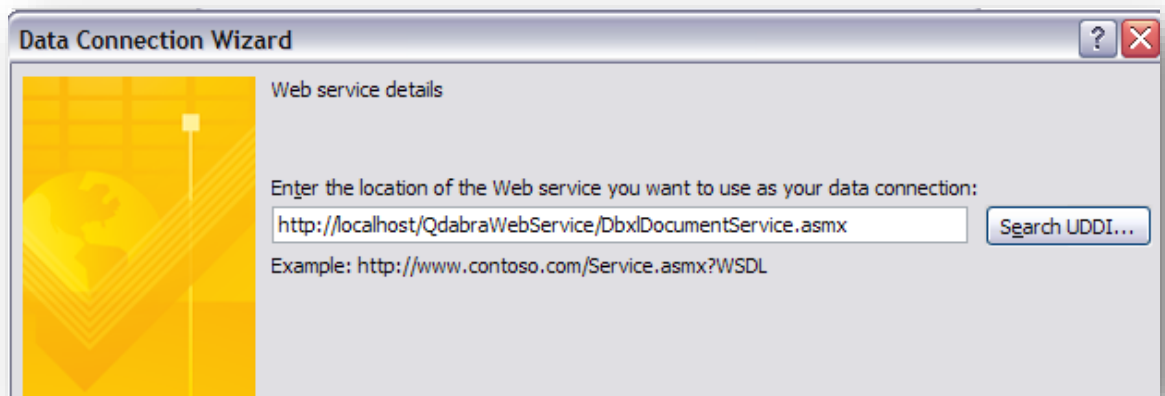


The data will be returned from a Web service:

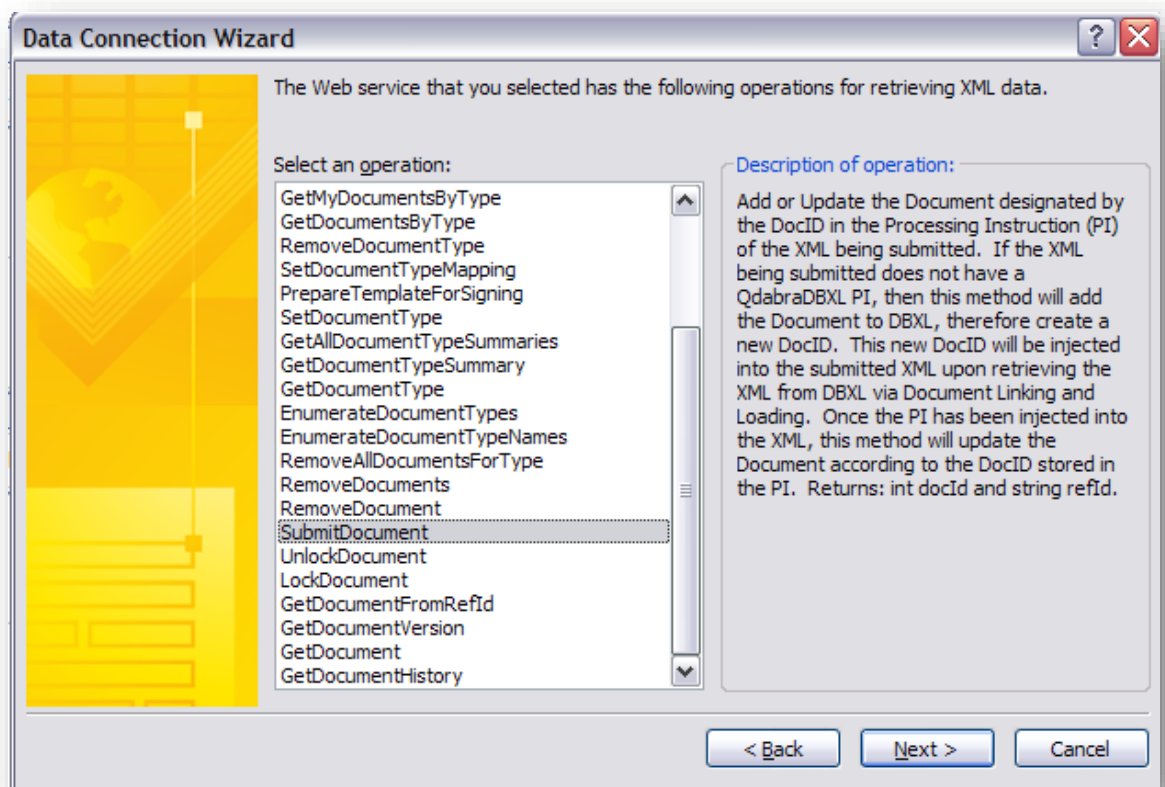


Use your DBXL QdabraWebService for the location of the web service, and select the DbxlDocumentService:





Select the SubmitDocument operation:



You do not need to set any values for this data connection. You may name it whatever you like – but you will need to know the name for use in the qRules commands. Also, make sure to uncheck the “Automatically retrieve data when form is opened” checkbox on the last page of the wizard.



INSTALL QdFILE

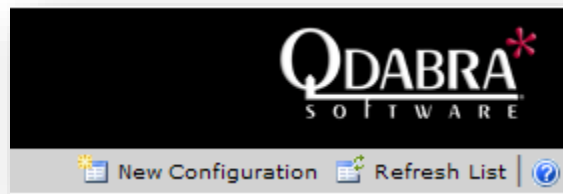
One of the qRules commands, *SaveToDbxl*, requires that a special DocType be created in **DAT** for use in saving pictures and attachments.

QdFile is already installed in system document type in DBXL ver. 2.4, However if you are using earlier version of DBXL, please follow the steps bellow.

A template for **QdFile** is copied into the installation folder when qRules is installed. The default path is *C:\Program Files\Qdabra Software\Qdabra Rules Library\QdFile.xsn*. Open **QdFile.xsn** in design mode and save it locally, and then publish the template to a network location.

To create a new DocType for the **QdFile** template, open **DAT** for your **DBXL** installation:

<http://<servername>/QdabraWebService/Forms/DAT/template.xsn> with <servername> replaced with the name of your server. Select **New Configuration**:



The DocType name should be **QdFile** and the published **QdFile.xsn** should be attached as the template:



General	Database	Permissions	Promoted
Document Type Details			
Name			
QdFile			
Description			
Last Modified By			
InfoPath Form Template			
XSN File			
 QdFile.xsn Microsoft Office InfoPath Form Template 4.04 KB			

Save the configuration to complete the installation of the DocType.

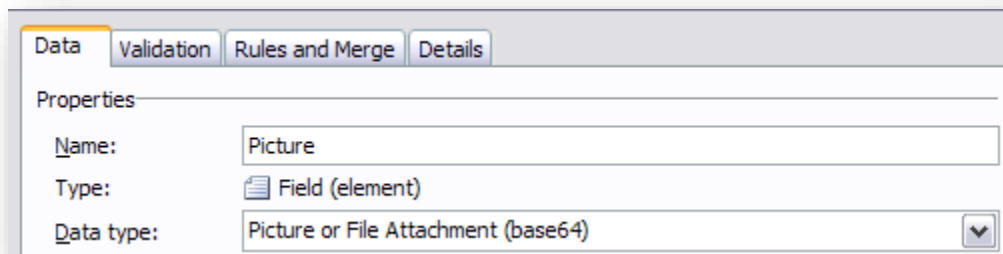


DATA STRUCTURE FOR PICTURE AND FILE ATTACHMENT NODES

Fields that will contain file or picture attachments that you would like to save to DBXL (QdFile) with the SaveToDbxl command or to SharePoint with the SaveToSharePoint command require special data structures.

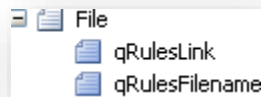
SaveToDbxl Command

The source field (which you wish to copy the picture or file from) should be a base64 data type:



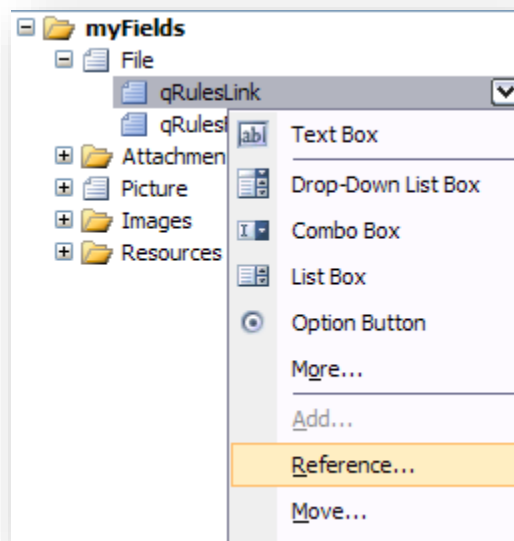
The screenshot shows the 'Data' tab of the Properties dialog box. The 'Name' field is set to 'Picture'. The 'Type' is set to 'Field (element)'. The 'Data type' is set to 'Picture or File Attachment (base64)'.

The field needs to have two attributes (string data type), **qRulesLink** and **qRulesFilename**:

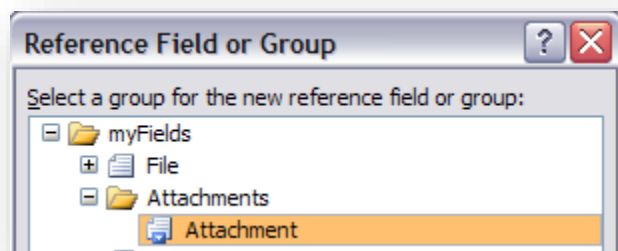


If you have more than one source field in a form, you will need to create a reference from your existing **qRulesLink** and **qRulesFilename** attributes to the new field. Right click the attribute and select **Reference**:

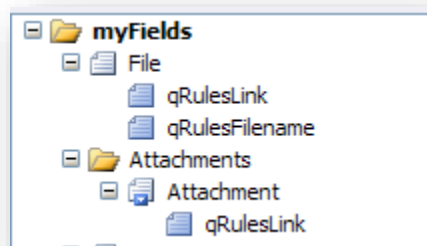




The **Reference Field or Group** dialog window opens. Select the new field you'd like to add the attribute to:



Select **Okay**. The attribute will have been added to the new field:

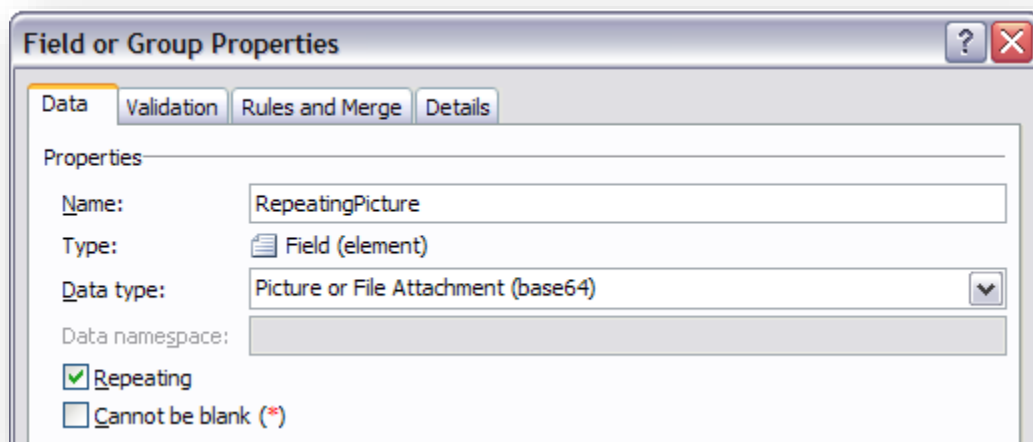


After your attachment has been submitted as a **QdFile** document to **DAT**, **qRulesLink** and **qRulesFilename** will be populated with the link and the filename to the image or attachment. You can use conditional formatting in your form to show or hide this information as well as the file or picture attachment control.

For example, if you choose to hide the attachment control if **qRulesLink** or **qRulesFilename** are not blank, and hide controls bound to **qRulesLink** or **qRulesFilename** if they are blank, the user will have the attachment control when they add a picture or a file, and a link that can be selected after the file has been transferred to **DAT**.

Saving multiple files/images

The source field (which you wish to copy the pictures or files from) should be a base64 **repeating** field:



The field needs to have two attributes (string data type), **qRulesLink** and **qRulesFilename**:



With this data structure as a sample, the value of the **Command** field in **QdabraRules** would be set to:

SaveToDbxl /submit=SubmitDocument /xpath=/my:myFields/my:Images/my:Image

Please note that in the multiple file commands, the XPath points to the repeating field within the group. You can also use a non-repeating field inside a repeating group.



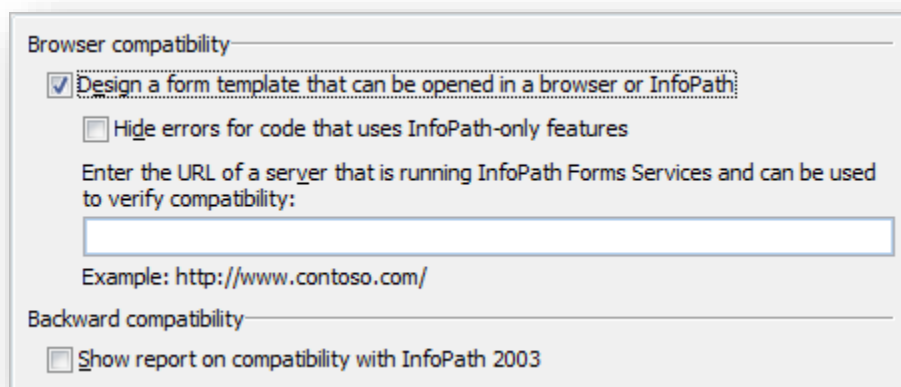
USING qRULES IN INFOPATH FORMS SERVICES (IPFS)

Many qRules commands are browser compatible, allowing you to use your qRules solutions in InfoPath Forms Services – the compatibility matrix in the Qdabra Rules Command Overview outlines any specific restrictions. Since qRules injects code into your forms, you will have to publish your XSN as an admin-approved form template. This allows you to manage the form in one place and use it in many places. This section details the steps on how to admin-deploy your qRules solutions to be able to use them in the browser. Please note that if you are using SharePoint 2010 or 2013, have sandboxing enabled, and are a site collection admin, you can also publish directly to a library.

Publishing in InfoPath 2007 and SharePoint 2007

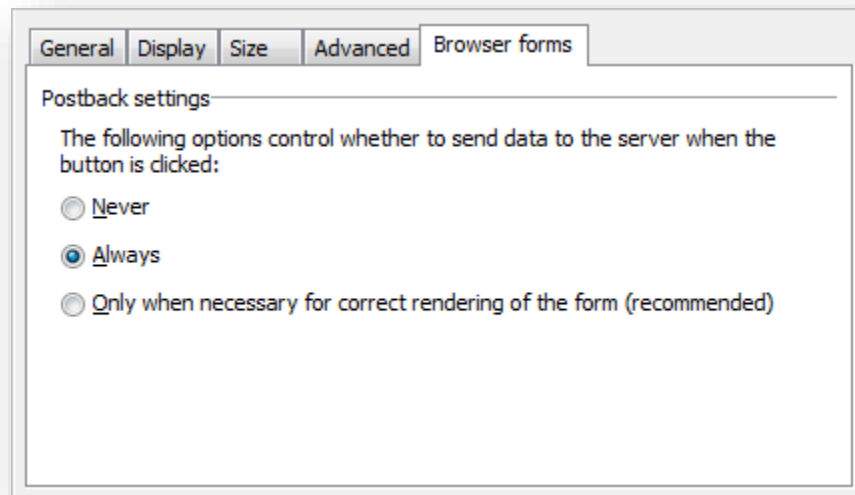
You can follow similar steps in SharePoint 2010 or SharePoint 2013 if you wish to admin deploy your form.

1. Go to **Tools > Form Options > Compatibility**, and enable browser-compatibility:

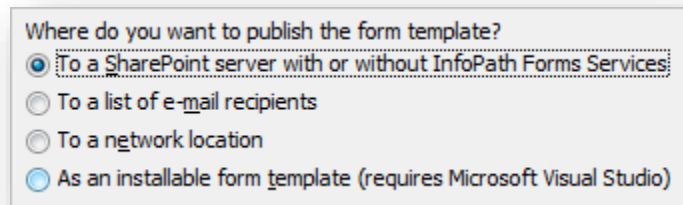


2. Make sure that any controls that make use the qRules commands are set to ALWAYS send data to the server. To do this, double-click on the control to show its properties, and go to the **Browser forms** tab, then select **Always**.



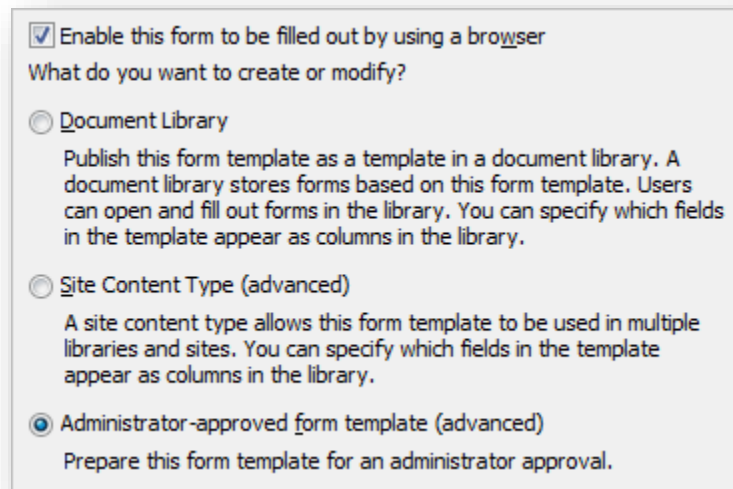


3. Save your form template to your network or local hard drive. For the purposes of this document, we will name our form template '*qRulesForms*'.
4. In InfoPath's menu, go to **File > Publish**.
5. Select the option to publish the form template 'To a SharePoint Server with or without InfoPath Forms Services' and click **Next**.



6. Enter the address of your SharePoint or IPFS site and click **Next**.
7. The next page will allow you to publish your form as an **Administrator Approved form template**. Make sure that option is selected and click **Next**.





☒ Enable this form to be filled out by using a browser

What do you want to create or modify?

☐ Document Library

Publish this form template as a template in a document library. A document library stores forms based on this form template. Users can open and fill out forms in the library. You can specify which fields in the template appear as columns in the library.

☐ Site Content Type (advanced)

A site content type allows this form template to be used in multiple libraries and sites. You can specify which fields in the template appear as columns in the library.

☒ Administrator-approved form template (advanced)

Prepare this form template for an administrator approval.

8. Browse to the local or network location where you wish to save your published form template to, click **Save** and then click **Next**.

9. Add some columns to be promoted to the SharePoint library and click **Next**, and then click **Publish**.

You will see a message that your form was published successfully and you are to give your SharePoint Administrator the information listed on the screen.

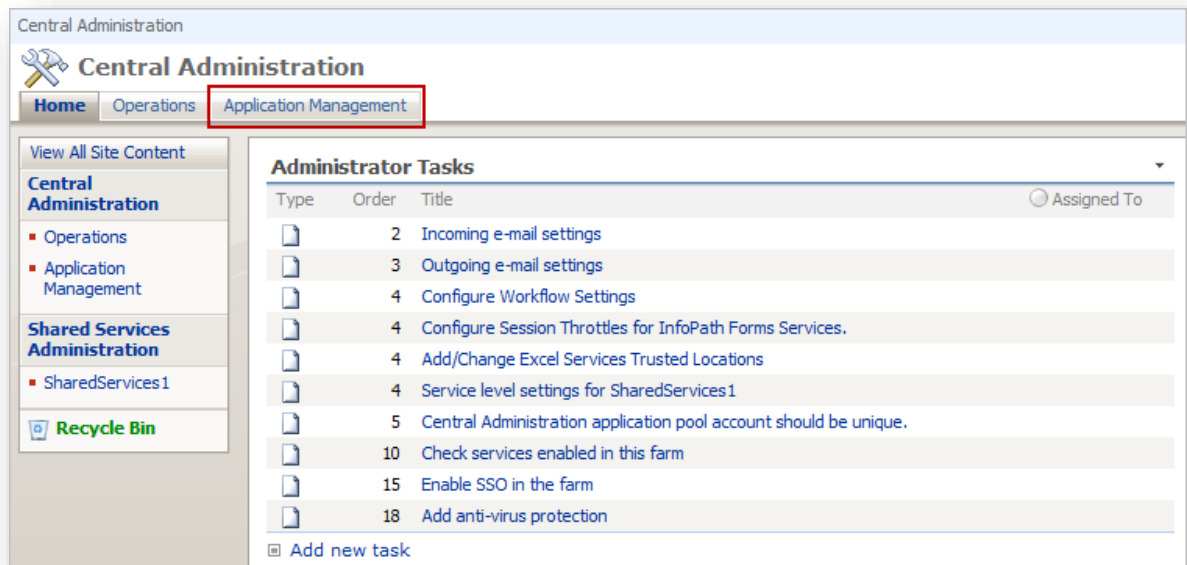
10. Click **Close** and close **InfoPath**.

You should be a SharePoint Administrator with access to SharePoint Central Administration to complete the next steps.

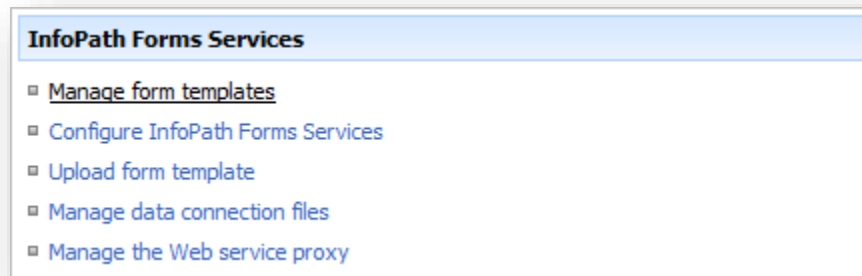
11. Navigate to the Central Administration web site.

12. Click on the **Application Management** tab.

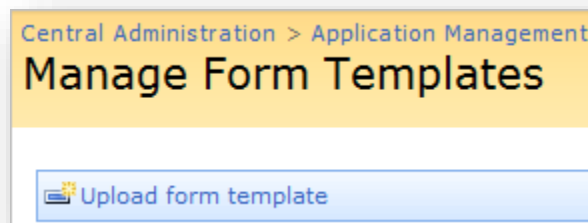




13. Under InfoPath Forms Services, click **Manage form templates**.



14. In the 'Manage Form Templates' page, click **Upload form template**.



15. In the 'Upload Form Template' page, browse to the location where you saved your published form template and click **Save**.



Central Administration > Application Management > Manage Form Templates > Upload Form Template

Upload Form Template

Upload Form Template

Browse to the form template you intend to add to the server.

File Name:

D:\qRulesForms.xsn

Before uploading, you can check this form template for detailed errors, warnings, and other information.

Upgrade

Set the behavior if the server contains a previous version of this form template with the same Form ID.

☒ Upgrade the form template if it already exists

New sessions will start using the upgraded version of the form template. Forms already open will continue to use the current version of the form template.

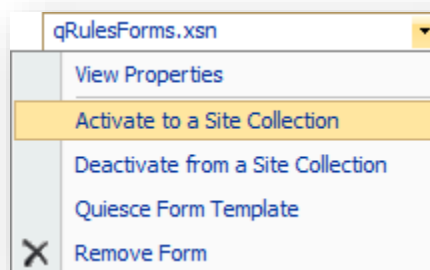
How would you like to handle existing browser-based form filling sessions?

☒ Allow existing browser-based form filling sessions to complete using the current version of the form template.

☐ Terminate existing browser-based form filling sessions. Any data in those sessions will be lost.

If you wish to wait until all sessions of the form template have completed before upgrading, navigate to Manage Form Templates, select the form template, and choose Quiesce form.

16. Accept the other defaults and click **Upload**. You should see a message indicating a successful upload. Click **OK**.
17. In the 'Manage Form Templates' page, click on the down arrow next to the template that you just uploaded and select **Activate to a site collection**.



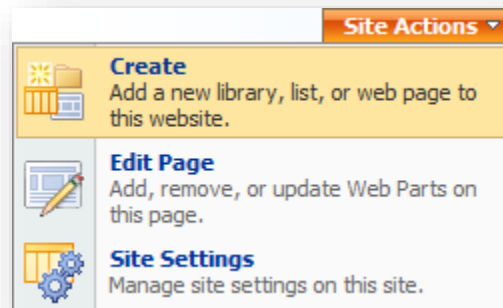
18. If the Activation Location Site Collection shows an incorrect site, click on the down arrow next to **Site collection** and select **Change Site Collection**. (Note: SharePoint almost always defaults to the *wrong* site collection the first time a form template is activated).
19. In the pop-up window, click the **Web Application** selector menu and choose **Change Web Application**.



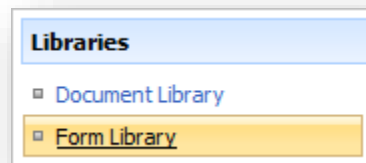
20. Click **SharePoint - 80**, and then click the **/** to activate to the entire site collection. Click **OK** twice.
21. Close Central Administration.

Create an IPFS-enabled SharePoint library

1. Navigate to the Site Collection in Internet Explorer.
2. Click **Site Actions > Create**

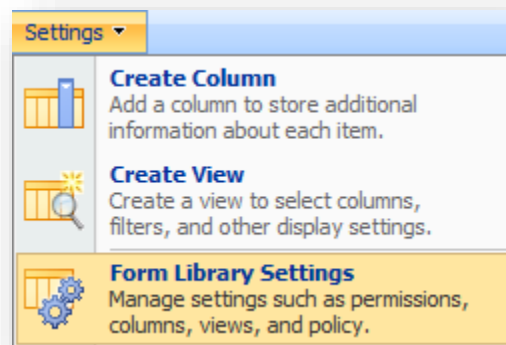


3. Click **Document** or **Form** library to create new library.

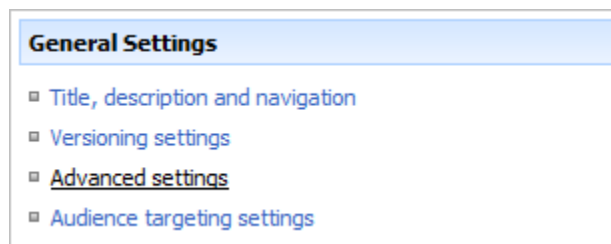


4. Continue to the page to create your new library. Give it a name of your choosing. For the purposes of this document, we will name our library '*qRulesForms*'. Select the options as desired, then click **Create**.
5. Go to library created and click **Settings > Form Library Settings**.

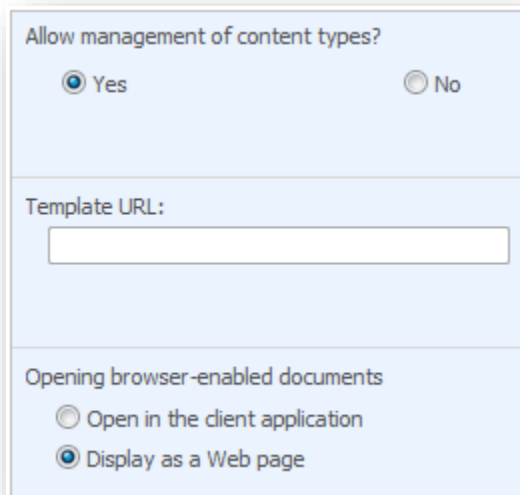




6. Click **Advanced settings** under *General Settings*.



7. In the Advanced Settings Page, select **Yes** under **Allow management of content types**, click **Display as a Web page** to open forms in the browser, then click **OK**.



8. Back on the Library Settings Page, you should now see a section called **Content Types**. Choose the link **Add from existing site content types**.



Content Types		
This document library is configured to allow multiple content types. Use content types to specify the information you want to display about an item, in addition to its policies, workflows, or other behavior. The following content types are currently available in this library:		
Content Type	Visible on New Button	Default Content Type
Form	✓	✓
☐ Add from existing site content types ☐ Change new button order and default content type		

9. Find your administrator-deployed template name in the **Available Site Content Types** list box and click **Add**, then click **OK**.
10. Back in Settings page, click **Change new button order and default content type** under Content Type section.

Content Type	Visible on New Button
Form	✓
qRulesForms	✓
☐ Add from existing site content types ☐ Change new button order and default content type	

11. Uncheck the Visible check box for Content Type 'Form' or 'Document' and click **OK**.

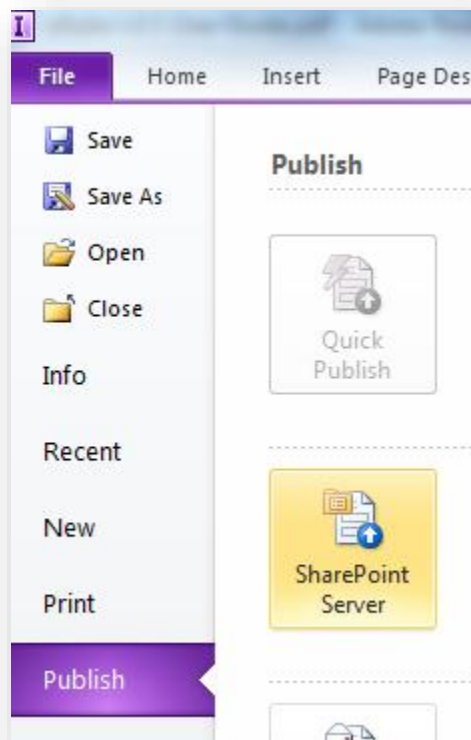
Now when users click the **New** button in the document library, your Administrator-Approved form will launch in the browser.

Publishing in InfoPath 2010 and SharePoint 2010

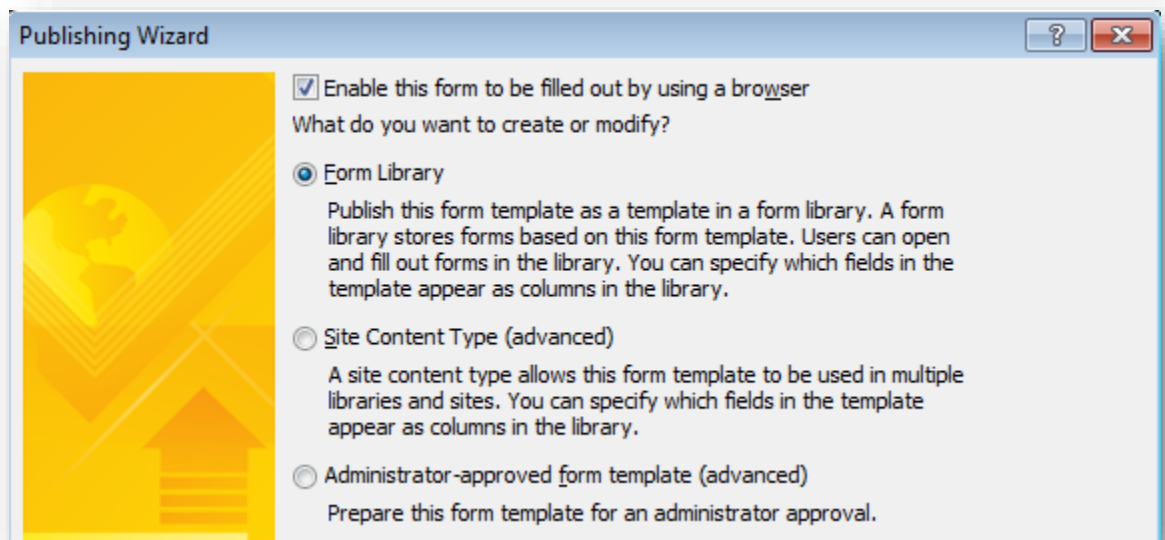
To publish an InfoPath 2010 form to SharePoint 2010, you can follow steps similar to those in the previous section (i.e. admin-deploy the form template). If you have sandbox permissions (that is, if you are a site collection administrator), you can simplify the process.

1. With the form open in Design mode in InfoPath 2010, go to **File > Publish > SharePoint server**.





2. Follow the Publishing Wizard to publish the form template to SharePoint 2010. You can publish directly to a Form Library and browser-enable the form.



Note that if you do NOT have sandbox permissions, the form will publish, but you'll get an error message when finishing the Publishing Wizard. However, the Design Checker will display no errors. In some cases, when using SharePoint 2010, your form might need to be admin-deployed. Contact us if you need assistance.

Publishing in InfoPath 2013 and SharePoint 2013

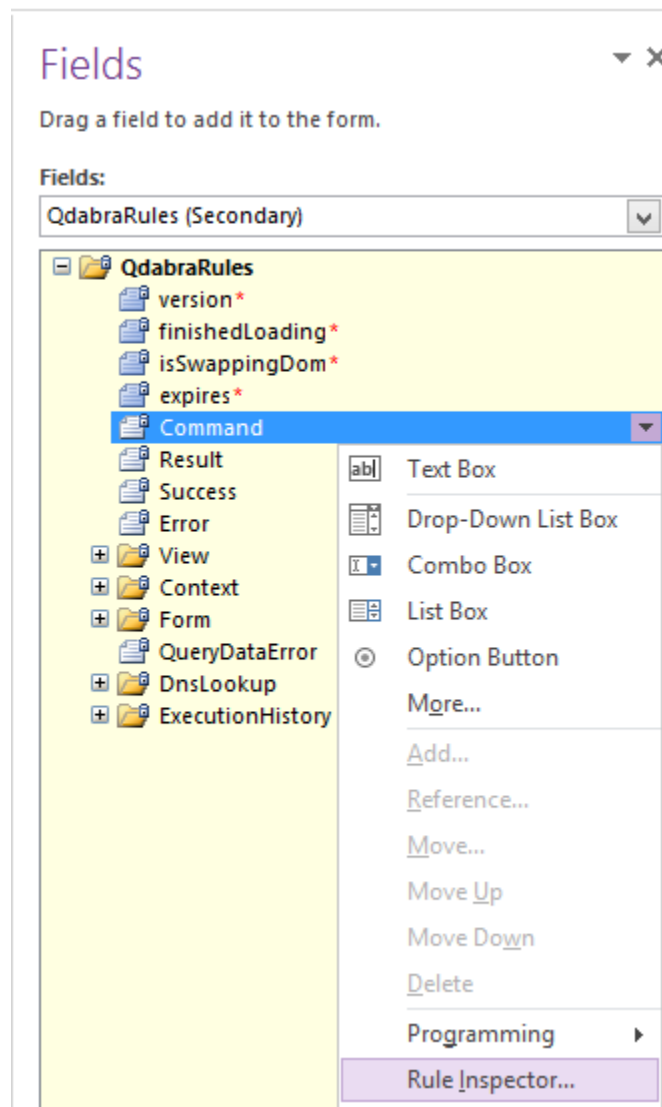
To publish an InfoPath 2013 form to SharePoint 2013, you can follow the procedure for Admin Approving your template as outlined in the [Publishing in InfoPath 2007 and SharePoint 2007](#) section, or you can follow the instructions in [Publishing in InfoPath 2010 and SharePoint 2010](#) to sandbox the form, if you have the appropriate permissions and sandboxing is enabled on your server.



REMOVE QRULES FROM A FORM TEMPLATE

The qRules injector in qRules 8.1 has been updated to uninject all qRules files, data connections and the code DLL. However, you will need to manually remove any rules executing qRules commands, as well as manually remove any controls from your form that are bound to qRules data sources.

1. Use the Rule Inspector to find any rules that are using the Command node from the QdabraRules data connection:

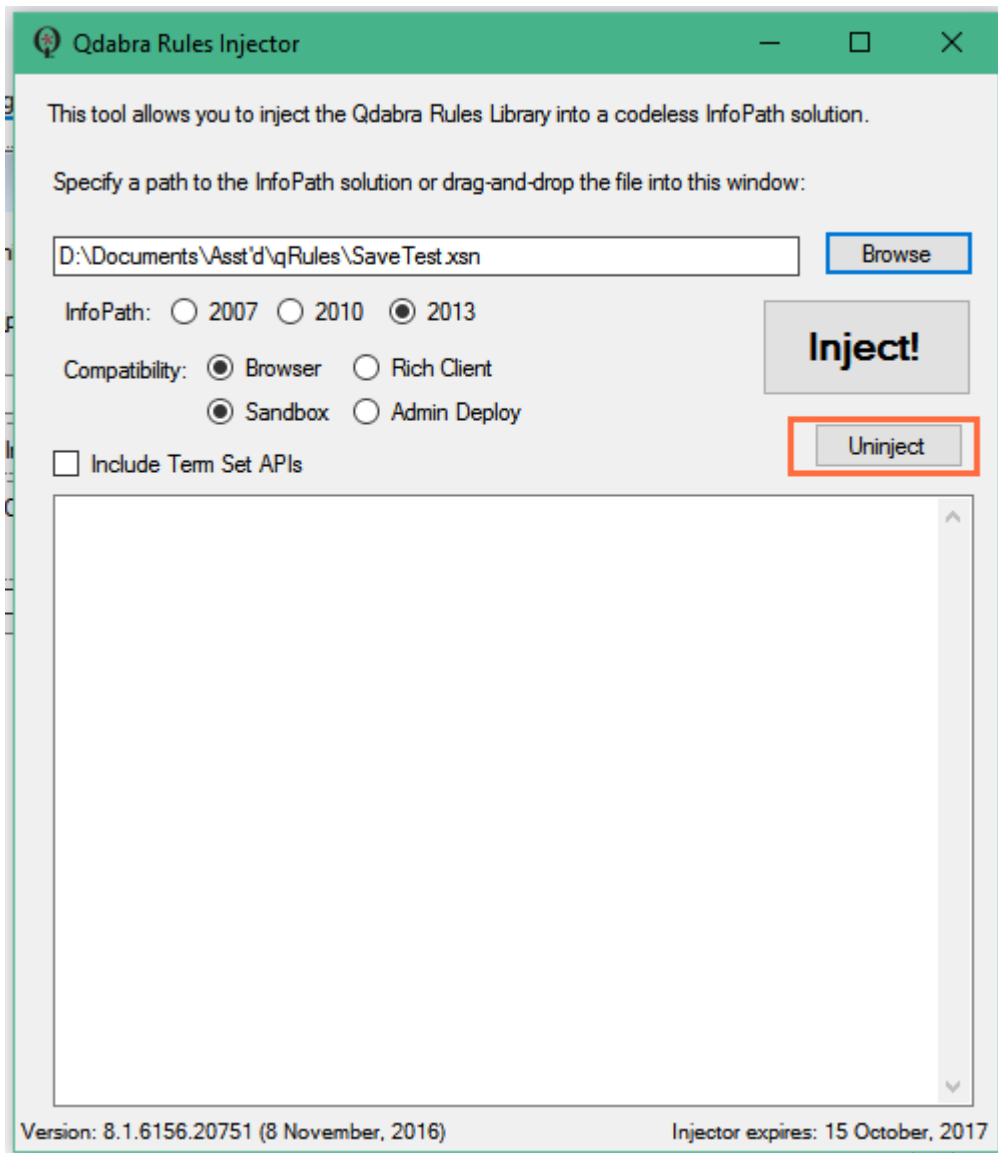


Make a note of the rules that set this field's value and go through your form and remove them.

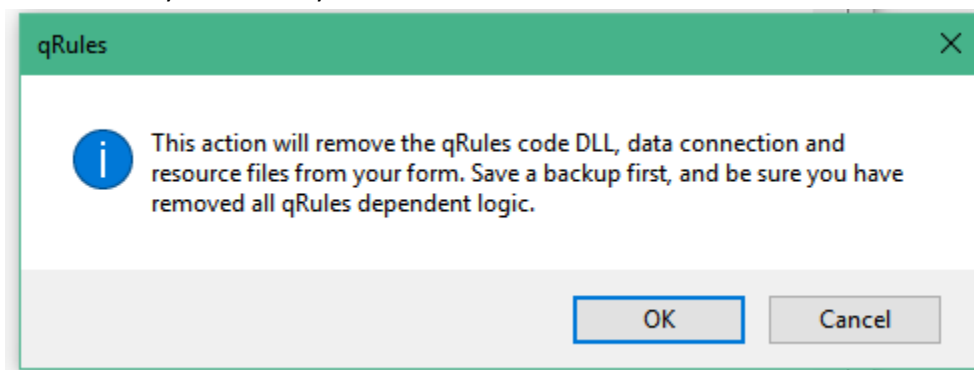
2. Check your form views, remove any controls that are bound to fields in the QdabraRules data source and save your form



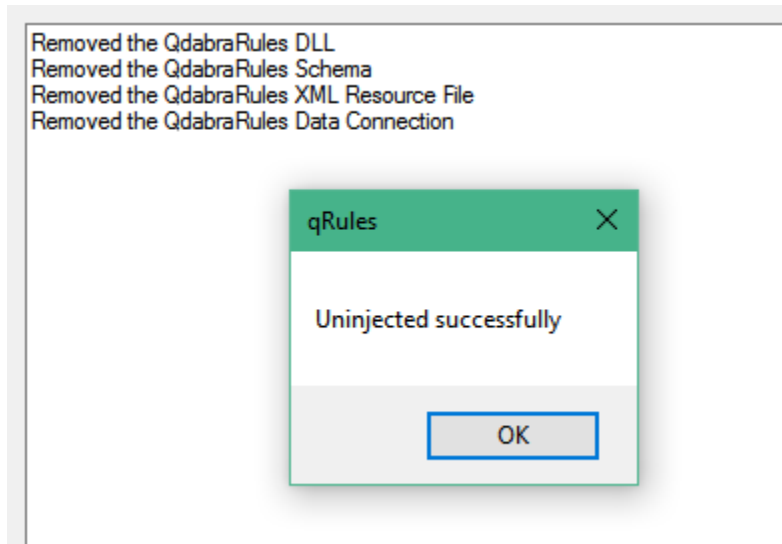
3. Open the qRules Injector and select the form from which you wish to remove qRules, then click Uninject:



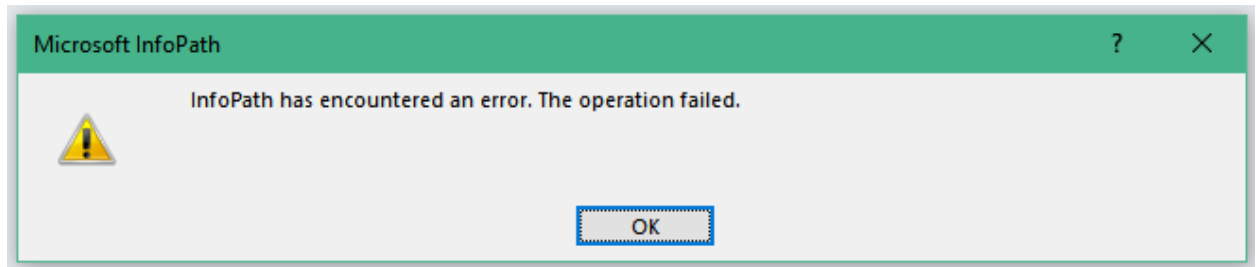
4. Confirm that you are ready to remove:



5. The injector will confirm the removal:



6. If you did not remove all controls on your form bound to fields in the QdabraRules data source, when you try to re-open your form in design mode, you may see an error:



Re-injecting the form should allow you to re-open your form in design mode so that you can finish removing qRules artifacts before reattempting the uninjection. If you have issues, we can help – contact support@qdabra.com



CHANGE LOG

These are the previous qRules versions:

- v1.1, v1.5, v1.6, v1.7, v1.8
- v2.0, v2.1, v2.2, v2.3, v2.4
- v3.1, v3.2, v3.3, v3.4
- v4.0, v4.1, v4.2, v4.3
- v5.1, v5.2
- v6.1
- v7.1, v7.2
- v8.1

The sections below detail the changes in each version.

qRules v8.1

- Updated commands
 - A number of commands were revised to return better error messages, including a better error message in the case a parameter is specified more than once.
 - DateDiff / DateAdd
 - Fixed a bug with excluding holidays if the start date includes a time.
 - DelimitedList
 - Added the option to use a blank space for a separator.
 - ExecuteSql
 - Added new timeout parameter.
 - Encode
 - Add new type option for parameter of sharePointFilename
 - SaveToSharePoint
 - Added new timeout parameter.
 - SubmitToDbxl
 - Add support for WithUserKey versions for SubmitDocument and SubmitDocumentWithKey
 - SubmitToSharePointList
 - Fixed a bug that might return an incorrect ID number to an item if some items submit successfully and others do not.
 - Fixed a bug that may cause attachments with special characters in their name to fail to save.
 - Added new parameter to allow a user to submit list items to a specific folder.
 - SwapDomWithDocument
 - Add support for WithUserKey version of web service
- Injector



- Added uninject button to remove qRules from a form.
- Injector expiration date is now displayed on the injector.
- Fixed a bug that occurred if the user had injected with TermSets and then injected again without.
- No longer needs to be installed as admin.
- Can now inject forms with special characters in the file name.

qRules v7.2

- New commands
 - GetViewInfo: Provides information about the view that is currently open in the form.
 - JsonToXml: Converts a JSON string to XML and places the converted XML into one of the form's data sources.
 - ListInputParameters: Provides a list of the available input parameters in the currently opened form, as a comma-separated list of all of their keys.
 - MakeRequest: Carries out a network request and provides the result.
 - XmlToJson: Converts XML in one of the form's data sources to a JSON string. Useful in combination with the MakeRequest command for interacting with JSON web APIs.
- Updated commands
 - RefreshSharePointListItems:
 - Fixed a bug that prevented items from being refreshed if qRulesLastModified is blank
 - Fixed a bug that prevented using SharePoint List data connections as the data source for the Refresh command.

qRules v7.1

- New commands
 - SendEmail: Augmented existing command to support sending e-mails through Outlook rather than through an e-mail data connection. This allows using files attached in the form as attachments on the e-mail.
 - GetConnectionURL: returns the current URL of the specified data connection.
- Updated commands
 - Encode: Fixed compatibility bug. Changes in InfoPath 2013 caused this command to not work in InfoPath 2013. This was fixed so that encoding/decoding URLs, URIs and URI Components works in IP 2013.
 - Transform: Now supports using xsl:include and xsl:import to import other XSLTs that are stored in the form as a resource file.
 - SubmitToSharePointList: Fixed error handling. Previously, the command would show a cryptic error if the /mappingName specified was not present in the mapping file. It should now provide an informative error.



- GetXML: /xpath parameter is now optional. If unspecified, the command returns the entire XML of the specified data source (or the main data source if /dsname is also unspecified)
- AssignDocument: Fixed regression preventing the use of the /myinfo parameter

qRules v6.1

- New commands
 - CreateSharePointFolder: allows creating a folder in a SharePoint list or library.
 - ExportView: uses InfoPath's built-in functionality to render the current view as a file and save it to a local drive.
- Updated commands
 - SaveToSharePoint: A parameter called /uniquename was added. It is a boolean parameter and it defaults to true if unspecified. When it is true, qRules will append the current date and time to the file name to ensure that it is unique. When false, qRules will use the actual file name, which can cause collisions if a file with that name already exists in the target location.
 - Encode: Added a new option for the type parameter in order to URI encode strings that are not actual URIs. It uses Uri.EscapeDataString:
<http://msdn.microsoft.com/en-us/library/system.uri.escapedatastring.aspx>.
 - SubmitToSharePointList and RefreshSharePointListItems: The commands have been updated to incorporate hierarchical list functionality, allowing relationships between parent and child lists.
- Mapping Tool
 - The Mapping Tool was updated to accomodate the changes made to the SubmitToSharePointList command. The UI has been updated to show parent-child relationships, and a Parent List ID option has been added.
- Injector
 - If a user modifies the QdabraRules data connection, the form creates a QdabraRules1.xsd file. On a later injection, the injector adds QdabraRules.xsd again. The Injector was updated to address this issue and avoid creating the extra XSD file.
 - The injector was updated to prevent injection of forms that contain custom code.
- Other updates
 - Due to a Microsoft update, InfoPath was crashing whenever a field with a default value referencing a People Picker executed a qRules command. A fix was implemented in qRules to avoid this crash.
 - Execution History repeating node was not displaying the entire Command. qRules was fixed to include the entire Command in this history.

qRules v5.2

- New commands



- GetListGuid
 - LookupDns
 - SetXml
- Updated commands
 - Transform: Added new optional parameters called /placement and /usedocument.
 - ChangeConnectionUrl now allows changing a SharePoint List data connection (2010 and later).
 - SubmitToSharePointList was updated to:
 - No longer encode ampersands and other special characters
 - Improve handling of rich text values mapped to SharePoint columns
 - SaveToSharePoint updated to remove illegal characters from the filename prior to uploading.
 - SubmitToDbxl now allows using both /user and /xpath parameters simultaneously.
- Other updates
 - Added the ExecutionHistory section in the QdabraRules secondary data source, to keep track of all Commands executed, along with the Result, Success and Error of each one.

qRules v5.1

qRules v5.1 includes compatibility with InfoPath 2013, SharePoint 2013 and Windows 8.

- New commands
 - ConvertTimeZone
 - Copy
 - EncodeBase64
 - GetErrors
 - GetTimeZone
 - GetUserName
 - IsDestReachable
 - Move
 - SetAutoUpdate
 - ShowDialog
- Misc fixes
 - Fixed an issue in the Mapping Tool to correctly render the schema of forms when the form template does not use "my" namespace prefix.
 - Improved the Start Menu folder structure to include documentation.
- Updated commands



- CopyTable returns a more informative error when source and destination tables do not match.
- SubmitToSharePointList:
 - Fixed issue when special characters were mapped and incorrectly encoded.
 - If the mapping does not generate any rows to submit to SharePoint (in the case of using filters in the mapping file), the command will still return Success. However, if you have a mapping error like a bad xpath, you should see an informative error.
- SwapDomWithDocument now allows a version parameter to load a previous version of a form that is stored in DBXL.
- SaveToSharePoint now correctly uploads file attachments that contain a period (dot) in their filename.
- Encrypt and Decrypt have been updated so that the clear parameter accepts true/false and 1/0 in addition to yes/no.
- RenderForm has been updated to fix some issues when rendering forms, such as:
 - Textboxes were rendering as fixed-height textboxes that could not be scrolled to see all of the text. They should now expand to show all text in the rendered HTML.
 - There was an issue where having XML characters (&, <, etc.) could cause the RenderForm command to fail.
 - Rendered forms were not showing text that had been manually typed into ComboBoxes. This should be fixed in this checkin
- OpenForm has been updated to include a parameter that allows opening existing XML forms for editing.
- AttachToSharePointList, SaveToSharePoint and SubmitToSharePointList have been updated to include a parameter (/clear) that allows the user to control whether the attachment is cleared from the main data source once the upload had finished.
- AttachToSharePointList has been improved:
 - Allow a SharePoint List Name in the /guid parameter.
 - Fix an issue when using the listName in the AddAttachment data connection, if it was provided in the data connection wizard.

qRules v4.3

- Fixes
 - qRules forms used to be installed via an install.js script. This has been replaced with FormsInstaller.
- New commands
 - AlterImage
 - FormatNumber
 - GetFormProperty
 - OpenForm



- RenderForm
- SubmitAsUser
- SubmitToSharePoint
- Updated commands
 - In previous versions of qRules, a **SharePoint list mapping** would use the list GUID. Starting in v4.3, the mapping can use the list name instead, which makes moving the form easier.
 - Previously, **CopyTable** would sometimes discard the nillable attribute, making a field required. This has been fixed in v4.3 so CopyTable brings in the nil attribute.
 - **SaveToSharePoint** now has a parameter called `namexpath`, an attribute relative to the attachment, which provides the name for the file being uploaded.
 - **SwapDomWithDocument** now has a parameter called `/upgradedocument` which allows upgrading the xml to upgrade to the latest template.
 - **GetValue** has a new parameter called `/eval`. Previously, the GetValue command could only be used to get the value of an actual node. Now it can also be used to get the value of an XPath expression, as in `concat(/group/group[1]/field, ' ', /group/group[2]/field)`. When `/eval` is true, it treats the `/xpath` parameter as an expression. When it's not true, it uses the old approach.
 - The **Transform** command has a new parameter called `/excluderoot`. When `/excluderoot` is true, the command will insert all of the XML *beneath* the root of the transformed XML into the destination node. When it is false, the entire transformed XML is inserted into the destination node.
 - Previously, the **Transform** command only allowed transforming all or part of one of the form's DOMs (main or secondary), but the `/sourcefile` parameter allows applying a transform to one of the template's files (a resource file or any other XML file in the template). The value of `/sourcefile` would be the name of the file to transform.
 - Implemented iterative functionality for **ReplaceString**. `/reldest` is an XPath that is relative to the value of the `/xpathsrc` parameter, and when `/reldest` is specified, qRules will iterate through all nodes that match `/xpathsrc`, perform the string replacement, and place the result in the field indicated by `/reldest`. In the end, the number of iterations performed is placed in the qRules Result field. `/append` is a boolean value and when it is true, the result of the string replacement is appended to the field indicated by `/reldest` instead of overwriting it.
 - **SubmitToDbxl** now allows submitting via other web methods. Previously, only SubmitDocument was allowed. Now, SubmitDocumentAsUser can be used (along with the new parameter `/user`) and SubmitDocumentWithKey can be used (with the `/key` parameter).
 - Fixed an issue with **RefreshSharePointListItems** when refreshing rich text list columns that contained blank lines.
 - RefreshSharePointListItems now contains a parameter called `/xpathsrc`. The `xpathsrc` refers to the data connection specified by `dsname`, a SharePoint List data connection to



your list. You have to have already performed the query to get the items - this way you can provide whatever filtering you need to make sure you have the right items available for refreshing. When you create your SharePoint List data connection, you have to include the Modified field (we use that to know which is newer / older between the form and the list).

- **SubmitToSharePointList** allows including an attachment in the list item. When mapping more than one attachment, the new namexpath allows specifying the filename that will be used for each attachment. The namexpath parameter will take a relative xpath from the Attachment field.

qRules v4.2

- New universal functionality
 - Any command that returns data to the QdabraRules Result node can now also return data to a field of your choice with the use of the /resultdestds and /resultxpath parameters
 - **resultdestds** (optional): data source name if /resultxpath is not in the Main data source
 - **resultxpath** (optional): XPath to field for result

This means that you no longer need to set rules to move information from the result node into your form fields.

- New commands
 - File
 - DecodeBase64
 - Financial
 - Ddb
 - Fv
 - Ipmt
 - Irr
 - Mirr
 - Nper
 - Npv
 - Pmt
 - Ppmt
 - Pv
 - Rate
 - Sln
 - Syd
 - Math
 - Abs
 - Acos



- Asin
 - Atan
 - Atan2
 - BigMul
 - Ceiling
 - Cos
 - Cosh
 - DivRem
 - Exp
 - Floor
 - Log
 - Pow
 - Round
 - Sin
 - Sinh
 - Sqrt
 - Tan
 - Tanh
- String
 - CompareStrings
 - GetIndex
 - PadString
 - RemoveFromString
 - ReplaceString
 - SetCase
 - TrimString
- Fixes
 - Views group in QdabraRules 2DS now allows to turn on/off the view switch tracking.
 - Update to qRules logic for determining if SharePoint commands should be using the SharePoint object model or web services for greater accuracy.
 - Add useBrowserApi node to QdabraRules 2DS to allow override of qRules logic for determining the use of the SharePoint object model.
 - Installer forces users to grant admin permissions to setup process.
- Updated Commands
 - SaveToSharePoint
 - Now contains an /async parameter for asynchronous upload.
 - Performance improved for document upload.
 - RefreshSharePointListItems: improved messages to count items that match.
 - SubmitToSharePointList
 - You can now submit to multiple mappings using only one command.



- Mapping tool allows you to create multiple mappings in one mapping file.
- Mapping tool allows you to import mapping.xml for further edits.
- CopyTable: Added /xpathdest parameter in case the tabledest has a filter.
- SaveToDbxl now has a new parameter called /qdimage to allow uploading images.
- DateDiff and DateAdd can now exclude holidays when calculating weekdays.

qRules v4.1

- New commands
 - ListDirectory
 - ValidateUser
 - GetTermPath
 - GetTermSetIds
 - GetUserProfileByName
- Fixes
 - SharePoint related commands (SaveToSharePoint, SubmitToSharePointList, DeleteFromSharePoint, AttachToSharePointList, RefreshSharePointListItems) have been updated such that they call the SharePoint Object Model. This allows the commands to work in browser forms on Office 365.
 - Added a new node in the QdabraRules secondary data source (**/QdabraRules/Form/@useWebServices**) which should be set to **true** should the user wish to skip using the SharePoint Object Model. In essence, setting this node to **true** will make sure your form is not using the fix directly above this one and qRules will behave as it did in qRules v4.0 and earlier.
 - DateDiff and DateAdd now allow excluding week end days.
 - Minor fixes to the InfoPath to SharePoint List Tool to enhance compatibility with Office 365.
 - Fixed bug where reinjection did not add secondary data source necessary for TermSets feature.
 - Fixed minor bugs with injection of XTPs and upgrade (reinjection) of templates.
 - A more informative error message was added to CopyTable in certain scenarios with deep tables in the schema.
 - Added a new node (**/QdabraRules/Form/@extendedErrorMessages**) to the QdabraRules data connection. This is a Boolean (true/false) node. When set to true, the command will return additional error information, if available.
- Updated Commands
 - DateAdd and DateDiff now have the ability to exclude weekend days in date calculations – see the individual commands for additional information.

qRules v4.0

- New commands



- GetTermSets
- GetTermsByLabel

qRules v3.4

- New commands
 - SaveToPath
 - CreateDirectory
 - DeleteFile
 - ChangeConnectionUrl
- Fixes
 - Fixed SetValue so using the /nil option is browser-compatible.
 - Fixed behavior for nillable nodes added via the Insert command.
 - Improved exception messages returned by SaveToSharePoint.
 - Insert command is now iterative; see information below about the new /firstparentonly parameter.
 - Insert command fixed to present correct error message when /child value begins with a slash.
 - Fixed injector so forms injected with qRules can be used in Qdabra's QueryBuilder.
 - Fixed qRules so a Refresh button that refreshes all data connections will no longer cause an exception.

qRules v3.3

- Installer updated to resolve common installation issues.
- Injector updated to resolve an issue with InfoPath changing the location of the QdabraRules.xml.
 - In InfoPath 2007 forms with XML data connections where the connection wizard was walked through more than once and the XML is not included as a resource prior to creating the connection, InfoPath could generate form files that end in "." without an extension. This could cause the form to fail to open after re-injection with qRules. The qRules v3.3 injector avoids this issue by removing the trailing dot from the file reference in the manifest.
- New commands
 - GetFileName
 - ModifySqlQuery
 - OpenUrlInBrowser
 - Transform
 - ResetDbxIVersion
 - ExecuteSqlQuery
- Updated commands:
 - SubmitToSharePoint now allows a user to add an xpath filter to the repeating group they are submitting.



- SaveToSharePoint now has the ability to specify filename.

qRules v3.2

- Now Office365 compatible
- New commands:
 - Encode
 - GetDayOfYear
 - GetValue
- Updated commands:
 - Insert
 - Updated to include /before parameter for specifying node to insert before.
- New features:
 - FormUri added to the QdabraRules data source to return URI of template
 - IsNew and IsReadOnly added to the QdabraRules data source so commands are no longer required for this information.

qRules v3.1

- New commands:
 - Relink
 - SetFocus
 - DeleteFromSharePoint
- Updated commands:
 - AssignDocument
 - Updated the AssignDocument command to work with modified DBXL v2.5 notifications structure.
- New feature
 - Context node is added to QdabraRules to identify which node currently has focus
- Fixes
 - ChangeSubmitUrl no longer errors out if href (URL) has "Forms" in lower case.
 - Bug in SubmitToDbxl was fixed such that blank values for the DBXL tokens (Name, Author Description) no longer cause errors.
 - Bug in SubmitToSharePointList and RefreshSharePointListItems was fixed in case multiple items are specified.

qRules v2.4

- Installation and injection changes:
 - Installer now works on both 32 bit and 64 bit machines.
 - Installer now installs qRules Mapping Tool (formerly the InfoPath to SharePoint List tool) and the qRules Sample form.



- Injector now allows specifying between 2007 or 2010 InfoPath forms.
- New commands:
 - Print
 - DateAdd
 - GetXml
 - QueryData
- Updated commands:
 - Replace
 - Updated the Replace command with a new param (/nsredef) that will allow users to specify a namespace prefix to have the source definition replaced with the destination definition.
 - SubmitToDbxl
 - Allow submitting repeating subtree.
 - Allow submit with Key.
 - Use XPath to get doctype, id, key, name, author, desc.
 - Allow submit to set result docid to a field of choice.
 - DBXL Tokens can now be rich text.
 - SubmitToSharePointList
 - Updated to support forms with multiple namespaces in Main data source.
 - Command now strips any additional slashes in the URLs.
 - Improved error reporting.
 - SetValue
 - Now has a dsname parameter to set fields in a secondary data source.
 - Added /blank and /nil parameters
 - CopyRichText
 - Fixed to handle InnerXML
 - FilterOwssvr
 - Added a single character, optional delimiter, in case filter values have commas.

qRules v2.3

- New commands
 - AttachToSharePointList
 - GenerateRandomNumber
 - GetEnvironment
 - RefreshSharePointListItems
 - GetFullDateTime
 - SetDefaultView
 - SetCaption (client only)
 - SetSave (client only)
- Updates



- SetValue: Ability to include 2DS
- SubmitToSharePointList
 - New parameters added to include attachments (single and repeating).
 - Includes all the options available for AttachToSharePointList.
 - Now supports submission of rich text to a SharePoint List rich text columns (mapping must be created with updated IPToShP List tool)
- SaveToSharePoint – new option to provide data source to SharePoint's Copy Web Service to allow saving images or files to a different server than where the form resides (no cross domain issues)
- Copy Table – fix to prevent the view from being left in a non-updatable state.
- IPToShP List tool – adds ability to mark a field as rich text for use in submit to SharePoint list OR when using the tool as stand alone
- Injector – new ability to choose if you want browser or client compatibility.
 - If you choose client, you can also use SetSave & SetCaption.
 - Supports re-injecting and updating from client to browser.

qRules v2.2

- New commands:
 - AssignDocument
 - InsertPi
 - DeletePi
 - GetAttachmentAttribute
 - ClearErrors
 - DelimitedList
- Fixes:
 - qRules v2.2 can coexist with custom code
 - SubmitToSharePointList now accepts special characters

qRules v2.1

- New commands:
 - Encrypt
 - Decrypt
 - GetWeekDay
 - FormatDate
 - SwapDomWithDocument
- Changes in existing commands:
 - SaveToSharePoint is now browser compatible.
 - ChangeSubmitUrl now works without specifying the submit URL in browser-enabled forms that have been admin-deployed



- Specifying a doctype in SubmitToDbxl has been re-enabled for more robust retrieval of input parameters
 - CopyTable now allows copying of rich text fields within tables
- Injector changes:
 - Now warns users to backup form templates before injecting qRules
 - Injector now allows injection into template parts (XTPs)

qRules v2.0

- New commands:
 - SubmitToSharePointList
 - ChangeSubmitUrl
 - SetValue
 - FilterOwssvr
 - CopyRichText
- Changes in existing commands:
 - SaveToSharePoint now returns the actual attachment filename instead of a link

qRules v1.8

- Now browser-compatible!
- New commands:
 - IsNew
 - IsReadOnly
- Changes in existing commands:
 - Insert command now allows inserting before or after
 - Added Error and Success nodes to Qrules schema. Error messages are now being returned in the **Error** node instead of message boxes.
 - Modified error checking for validating secondary data sources so that we are able to return our custom error message (previous method caused unhandled exception)
 - Improved error handling

qRules v1.7

- New commands:
 - IsDirty
 - SetDirty
 - GetInputParameter
 - DateDiff
- Changes in existing commands:



- CopyTable and MoveTable now have the following parameters: tablesrc, rowsrc, tbldest and rowdest. Rows are relative to the tables.
- Boolean now allows 1/0, True/False and yes/no values. Previously it did not allow all of these. However, note that any value other than 1, True or Yes will be interpreted as a 0 (False, No).
- SubmitToDbxl no longer requires a DocTypeName parameter.
- Bug fixes:
 - CopyTable now allows copying of attributes.
 - Fixed CopyTable issue with rich textbox but only for when the rich text field is a child node of the row node. The row node is expecting to be text or a group, not rich text.
 - XSN (caps) file extension now allowed.

qRules v1.6

- New commands:
 - Replace
 - Insert
- CopyTable and MoveTable now have two additional parameters: tablesrc and tabledes.

qRules v1.5

- New command: MoveTable
- Changes in the formatting of commands
- Changed secondary data connection name from QdabraCommands to QdabraRules
- Schema changes in QdFile
- SubmitToDbxl now stores Name, Author and Description in XPath
- Bug fixes

qRules v1.1

- Initial public release

SUPPORT

If you have questions about the information in this document, please contact Qdabra Software for assistance. The qRules product page on [Qdabra.com](http://www.qdabra.com) will contain an updated list of all available documentation.

Licensed customers can contact us via Support@Qdabra.com. You can also use the InfoPathDev.com [Qdabra Product support forums](#) to request help from the community.



Contact Information

Qdabra Software
218 Main Street, Suite 731, Kirkland, WA 98033

Phone: 877.544.2389
Email: Support@Qdabra.com
Website: <http://www.qdabra.com>
Community: <http://www.infopathdev.com>

